

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 004.942

До захисту допущено
В. о. завідувача кафедри ММСА
О.Л.Тимошук
«__» _____ 2020 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз
на тему: «Система для створення відеорекомендацій на основі графових
нейромереж»

Виконала:
студентка II курсу, групи КА-91 мп
Бойчук Олена Андріївна

Керівник: доцент кафедри ММСА
к.т.н, доц. Дідковська М.В.

Рецензент: доцент кафедри програмного
забезпечення комп'ютерних систем
КПІ ім. Ігоря Сікорського, к.т.н., доц. Заболотня Т.М.

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших авторів
без відповідних посилань

Студент _____

Київ
2020

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)
Спеціальність — 124 «Системний аналіз»

ЗАТВЕРДЖУЮ
В. о. завідувача кафедри ММСА
О. Л. Тимошук
«__» _____ 2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту Бойчук Олені Андріївні

1. Тема дисертації: «Система для створення відеорекомендацій на основі графових нейромереж», науковий керівник дисертації Дідковська Марина Віталіївна, к.т.н., доцент, затверджені наказом по університету від «02» листопада № 3182-с.

2. Термін подання студентом дисертації: 14 грудня 2020 р.

3. Об'єкт дослідження: вхідні дані про матеріали відеоплатформи, показники взаємодії користувачів з відеоплатформою.

4. Предмет дослідження: рекомендаційні системи, алгоритм GraphSAGE.

5. Перелік завдань, які потрібно розробити:

- 1) Аналіз існуючих підходів для побудови рекомендаційних систем.
- 2) Збір, підготовка та аналіз даних для побудови моделі для прогнозування оцінок користувачів.
- 3) Аналіз результатів прогнозування моделі.
- 4) Побудова рекомендаційної системи на основі отриманої моделі для прогнозування оцінок користувачів.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- 1) Методи прогнозування, що застосовуються в рекомендаційних системах (2.1).
- 2) Модель GraphSAGE (2.2-2.4)
- 3) Вхідні дані (3.1 -3.7), побудова моделі (3.8-3.9)
- 4) Таблиці у розділі стартап-проекту

7. Дата видачі завдання: 01 вересня 2020 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації
1.	Концептуальний вступ дисертації. Формулювання об'єкта, предмета, цілі, завдань, новизни, практичної значущості результатів	01.09.2020 – 07.09.2020
2.	Перший розділ. Огляд літературно-інформаційних джерел.	08.09.2020 – 28.09.2020
3.	Другий розділ. Дослідження існуючих підходів до побудови рекомендаційних систем та алгоритму GraphSAGE.	15.09.2020 – 28.09.2020
4.	Третій розділ. Застосування дослідженого матеріалу на практиці, написання алгоритму обробки вхідних даних. Аналіз результатів роботи моделі для прогнозування оцінок користувачів, створення відеорекомендаційної системи.	29.09.2020 – 05.10.2020
5.	Четвертий розділ. Стартап-проект	06.10.2020 – 12.10.2020
6.	Концептуальні висновки. Перспективи розвитку отриманих рішень	13.10.2020 – 25.10.2019

Студент

О.А. Бойчук

Науковий керівник дисертації

М.В. Дідковська

РЕФЕРАТ

Магістерська дисертація: 71 сторінка, 4 розділи, 12 рисунків, 22 таблиці, 11 джерел.

РЕКОМЕНДАЦІЙНІ СИСТЕМИ, ВІДЕОПЛАТФОРМА, ПРОГНОЗ, НЕЙРОННІ МЕРЕЖІ, АЛГОРИТМ GRAPHSAGE.

Об'єктом дослідження є вхідні дані про матеріали відеоплатформи, показники результатів взаємодії користувачів відеоплатформи.

Предмет дослідження: прогноз оцінок користувачів відеоплатформи за допомогою нейромереж на основі графового представлення вхідних даних, побудова відеорекомендаційної системи для створення релевантного списку пропозицій для користувачів відеоплатформи.

Метою даної роботи є розробка робочої моделі для прогнозування оцінок користувачів за допомогою графового представлення вхідних даних про існуючі оцінки користувачів та інформації про матеріали відеоплатформи, створення відеорекомендаційної системи на основі моделі прогнозування оцінок.

Отримані результати: графове представлення вхідних даних про користувачів відеоплатформи, результати прогнозу оцінок користувачів відеоплатформи, відеорекомендаційна система на основі створеної моделі прогнозування оцінок. Програмна реалізація виконана за допомогою мови програмування Python.

ADSTRACT

Topic: ‘System for generating video video recommendations using graph neural networks’

Thesis: 71 pages, 4 sections, 12 figures, 22 tables, 11 sources.

RECOMMENDATION SYSTEM, VIDEOPLATFORM, FORECAST, NEURAL NETWORKS, ALGORITHM GRAPHSAGE.

The object of the study is the input data regarding video content at the video platform, as well as user-item interactions data.

Subject of research: link prediction of user-item interactions using graph input data representation, recommendation system for generating relevant user recommendations.

The purpose of this work is to develop a working model for predicting user-item interactions by using graph representation for input interactions and video content characteristics, the development of a video recommendation system using link prediction model.

The results obtained: graph representation of the input user-item interactions and content description, the results of a link prediction forecasts of the video platform users, video recommendation system based on the link prediction model. The software implementation is performed using the Python programming language.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	8
ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Вступ	11
1.2 Характеристики відеорекомендаційної системи	12
1.3 Зворотній зв'язок в рекомендаційних системах	15
1.4 Врахування контексту в рекомендаційних системах	17
1.5 Висновки до розділу	18
РОЗДІЛ 2. ДОСЛІДЖЕННЯ ПІДХОДІВ ДО ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ТА ОПИС МЕТОДУ ПРОГНОЗУВАННЯ ОЦІНОК КОРИСТУВАЧІВ	19
2.1 Вступ	19
2.2 Основні підходи до побудови рекомендаційних систем	20
2.3 Опис GraphSAGE для індуктивного представлення даних	28
2.4 Модифікація GraphSAGE для врахування гетерогенних даних	35
2.5 Висновки до розділу	37
РОЗДІЛ 3. АНАЛІЗ ВХІДНИХ ДАНИХ, ПОБУДОВА МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ ОЦІНОК КОРИСТУВАЧІВ ТА АРХІТЕКТУРА ВІДЕОРЕКОМЕНДАЦІЙНОЇ СИСТЕМИ	38
3.1 Вступ	38
3.2 Вибір платформи та мови програмування	38

3.3	Аналіз вхідних даних	40
3.4	Побудова моделі	46
3.5	Аналіз результатів прогнозу	49
3.6	Архітектура відеорекомендаційної системи	51
3.7	Висновки до розділу	53
РОЗДІЛ 4. СТАРТАП-ПРОЕКТ «VIDEO RECOMMENDATION SYSTEM»		54
4.1	Вступ	54
4.2	Опис ідеї проекту	54
4.3	Технологічний аудит ідеї проекту	57
4.4	Аналіз ринкових можливостей запуску стартап-проекту	57
4.5	Розроблення ринкової стратегії проекту	65
4.6	Розроблення маркетингової програми стартап-проекту	67
4.7	Висновки до розділу	70
ВИСНОВКИ		71
ПЕРЕЛІК ПОСИЛАНЬ		72

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

TF-IDF(англ. TF – term frequency, IDF – inverse document frequency) - статистичний показник, що використовується для оцінки важливості слів у контексті документа.

GCN(англ. Graph Convolutional Network) – графова згорткова мережа.

GNN(англ. Graph Neural Network) – графова нейронна мережа.

KPI(англ. Key Performance Indicator) - ключовий показник ефективності.

EDA(англ. Exploratory Data Analysis) - розвідувальний аналіз даних.

БД – база даних.

SGD (англ. stochastic gradient descent) – стохастичний градієнтний спуск.

ML (англ. Machine Learning) – машинне навчання.

MAE(англ. Mean Absolute Error) – середня абсолютна помилка.

MSE(англ. Mean Squared Error) – середня квадратична помилка.

RMSE (англ. Root Mean Squared Error) – корінь середнього квадрату помилки.

ВСТУП

Сучасні умови економічної конкуренції вимагають від підприємців створення все нових і нових технологій для залучення нових користувачів і збільшення часу їх взаємодії з веб-ресурсом. Однією з таких технологій є рекомендаційні системи. Дані системи привнесли нові способи взаємодії звичайних веб-сайтів зі своїми користувачами. Замість надання статистичної інформації, коли користувачі шукають за допомогою існуючої наперед заданої категорійної структури представлення доступного контенту, рекомендаційні системи збільшують ступінь інтерактивності, а також розширюють можливості, що надаються користувачеві.

Отже, в магістерській дисертації розглядається застосування графових нейромереж для прогнозування оцінок користувачів на веб платформі та побудова системи для створення відеорекомендацій на основі моделі для прогнозування.

Таким чином, метою роботи є розробка робочої моделі для прогнозування оцінок користувачів за допомогою графового представлення вхідних даних про існуючі оцінки користувачів та інформації про матеріали відеоплатформи, створення відеорекомендаційної системи на основі моделі прогнозування оцінок.

Для досягнення цих цілей виконані наступні задачі.

1. Проведено аналіз існуючих підходів для побудови рекомендаційних систем.
2. Проведений збір, підготовка та аналіз даних для побудови моделі для прогнозування оцінок користувачів.
3. Виконано аналіз результатів прогнозу моделі.

4. Розроблено архітектуру рекомендаційної системи на основі отриманої моделі для прогнозування оцінок користувачів.

Об'єктом дослідження є вхідні дані про матеріали відеоплатформи, показники результатів взаємодії користувачів відеоплатформи.

Предметом дослідження є прогноз оцінок користувачів відеоплатформи за допомогою нейромереж на основі графового представлення вхідних даних, рекомендаційної системи для відеоплатформи.

В якості методів дослідження використовуються гібридний підхід до побудови рекомендаційних систем і методи штучного інтелекту (модель GraphSAGE).

Наукова новизна отриманих результатів полягає в використанні алгоритмів з графовим представлення вхідних даних для відеорекомендаційної системи.

Практичними результатами роботи є розробка архітектури системи для створення відеорекомендацій, а також реалізація моделі для прогнозування оцінок користувачами відеоконтенту, що лежить в основі системи.

Робота складається з 4 розділів. В першому розділі аргументується актуальність роботи, проведений аналіз предмету дослідження. Другий розділ присвячений аналізу існуючих підходів до побудови рекомендаційних систем, наведено математичні основи моделі GraphSAGE, що використовується для прогнозування оцінок. У третьому розділі наведений опис практичної частини роботи, а саме аналіз доступних історичних даних, побудова моделі та аналіз отриманих результатів прогнозу. У заключному четвертому розділі описана стартап частина продукту, де проаналізовано можливі стратегії для виходу на ринок.

РОЗДІЛ 1. АНАЛІЗ ОСОБЛИВОСТЕЙ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Вступ

Для того щоб побудувати відеорекомендаційну систему, потрібно розуміти специфіку роботи відеоплатформи, структуру рекомендаційної системи, а також основні підходи до побудови та оцінки якості роботи рекомендаційної системи.

Рекомендаційні системи - один з найпопулярніших практичних застосувань здобутків інтелектуального аналізу даних та машинного навчання у сфері Інтернет-бізнесу. Рекомендаційна система аналізує поведінку користувачів Інтернет-сервісу, після чого може спрогнозувати оцінку користувачем того чи іншого об'єкту рекомендацій. Об'єктами рекомендацій можуть бути, наприклад, товари в інтернет-магазині, набори розділів на веб-сайтах, медіа-контент, а також інші користувачі веб-сервісу. На основі запропонованих рекомендаційною системою вподобань, поведінка веб-сервісу стосовно конкретного користувача може змінюватися та адаптуватись. Вхідними даними для моделей, що лежать в основі рекомендаційних систем можуть бути історичні дані про взаємодію користувача з платформою, інформація про контент, як-то опис фільму, режисер, актори, жанр, тривалість, інформація про користувача, наприклад, стать, вік.

Використання рекомендаційної системи на веб-ресурсі дозволяє підвищити цільові показники ефективності. Такими показниками можуть бути: обсяг переглянутого контенту, сумарний дохід від продажу контенту, перегляду реклами, тривалість користувацької сесії на сайті, ступінь залученості користувача, лояльність користувачів до веб сервісу.

В даному розділі будуть розглянуті основні поняття рекомендаційних систем, підходи до побудови відеорекомендаційних систем.

1.2 Характеристики відеорекомендаційної системи

Найбільш поширений серед користувачів Інтернету приклад персоналізації - персональні блоки видачі рекомендацій. На основі лише інформації про переглянуті користувачем об'єкти, або за використовуючи дані, надані користувачем про його вподобання, веб-сервіс ранжує об'єкти за зниженням показника релевантності для конкретного користувача.

Рекомендаційні системи допомагають користувачам орієнтуватися в великому обсязі контенту, розміщеного на сайті. У деяких випадках це необхідна функціональність. До прикладу, на сервісі “Youtube” розміщено записи, еквівалентом більше 100 років безперервного відеоконтенту. Користувач не може ознайомитись з кожним матеріалом, тому пошук відео, що йому потенційно може сподобатись, а також нового відеоконтенту, користувачі здійснюють, спираючись на рекомендації друзів чи на почуту інформацію у ЗМІ. У завдання рекомендаційної системи в даному випадку входить побудова персональної проранжованої стрічки з відеоконтентом, цікавим (релевантним) для користувача, на основі історії його взаємодії з відеопрогравачем та поведінкою на веб-сервісі загалом.

Рекомендаційні системи застосовуються не лише для рекомендацій користувачам найбільш релевантних об'єктів. Для збільшення ефективності промо-активностей, інтернет-магазини також використовують функціонал рекомендаційних систем для виявлення найбільш зацікавлених користувачів

у конкретному товарі чи послугі. На основі історії покупок користувачів, їх реакцій на вислані промо-листи, поведінку на веб-сайті інтернет-магазину, рекомендаційна система визначає ступінь зацікавленості кожного користувача в конкретному товарі, що дозволяє проводити промо активності стратифіковано для найбільш зацікавлених юзерів.

Отже, завдання рекомендаційних систем - запропонувати користувачеві товари чи об'єкти, про які він раніше не знав, але котрі можуть виявитися корисними чи цікавими для нього. Іншими словами, за допомогою цих систем знаходиться відповідь на питання про те, який продукт користувач забажає придбати чи переглянути в даний час. Рекомендаційні системи порівнюють однотипні дані від різних людей і розраховують список рекомендацій для конкретного користувача. Вони є альтернативою класичним пошуковим алгоритмам, оскільки дозволяють виявити об'єкти, які не можуть бути знайдені останніми[1].

Розглянемо основні характеристики, на базі яких можна описати рекомендаційну систему:

- предмет рекомендації (те, що рекомендується).

Предметом рекомендацій в сервісі може бути велике різноманіття об'єктів: товари(Amazon, Rozetka), новини(GoogleFeed, ukr.net), зображення(Behance, Pinterest), люди(Linkedin), музика(Deezer). В даній роботі предметом рекомендацій будуть виступати відео матеріали.

- Ціль рекомендації.

Наприклад, придбання товару, навчання, створення нових контактів. В даному випадку ціллю рекомендації буде надання користувачеві релевантних відео, тобто таких, котрі він перегляне після видачі.

- Контекст рекомендації (що робить користувач в даний момент)

Наприклад: переглядає список товарів, дивиться відео, спілкується з друзями.

- Джерело рекомендації.

Наприклад: експертна думка(коли мова йде про складний товар, як-то вино), аудиторія (середній рейтинг фільму на платформі), схожі за інтересами користувачі.

- Ступінь персоналізації.

Неперсональні рекомендації - рекомендації, що не враховують особистих переваг користувача. Тобто користувачеві рекомендуються об'єкти на основі регіону, часу. Більш ефективним є підхід, коли для генерації рекомендацій використовуються дані з поточної сесії користувача. Персональні рекомендації використовують всю доступну інформацію про користувача, в тому числі й історію його взаємодії з платформою.

- Прозорість.

Люди схильні довіряти рекомендації, якщо розуміють механізм її отримання. Таким чином ризик потрапити на недобросовісні сервіси, що рекомендують товари/контент за додаткову плату від продавця, або ж ранжують більш дорогі позиції вище знижується. Також можуть бути і ненавмисні маніпуляції. Наприклад, на фільм, що вийшов нещодавно, першими йдуть переглядати фанати, відповідно й рейтинг на нього в перший період буде сильно завищений.

- Формат рекомендацій.

Наприклад: спливаюче вікно, ранжований список, що міститься в певному розділі сайту, стрічка на екрані справа чи ще щось.

- Алгоритми, що лежать в основі системи.

Усі алгоритми, що використовуються зводяться до кількох базових підходів, які будуть розглянуті в наступному розділі.

1.3 Зворотній зв'язок в рекомендаційних системах

Зворотним зв'язком користувача на деякий об'єкт в рекомендаційних системах прийнято називати подію, за якою можна судити про ступінь вподобання об'єкта користувачеві. Ось кілька прикладів зворотного зв'язку від користувача:

- проставлення оцінки об'єкту за бальною шкалою ;
- натискання на кнопку “подобається” / “не подобається”;
- відвідування сторінки з описом об'єкта, перехід за посиланням на об'єкт (клік);
- відвідування сторінки з описом об'єкта більш ніж один раз (зацікавленість);
- додавання контенту в закладки .

Саме на основі зворотного зв'язку користувача на різні об'єкти, рекомендаційна система формує матрицю оцінок вподобань R , до якої потім застосовуються алгоритми генерації релевантних рекомендацій. Перетворення зворотного зв'язку в числове значення переваги - складне та дуже важливе завдання в налаштуванні рекомендаційних систем. Як правило, при виборі схеми оцінки переваги оптимізується метрика, безпосередньо

пов'язана з ключовими показниками ефективності (KPI) бізнесу. Техніки підбору схеми оцінки переваги виходять за рамки даної роботи.

За видами зворотного зв'язку, завдання моделювання переваги в рекомендаційних системах прийнято розділяти на два види:

- з явним зворотнім зв'язком (explicit feedback);
- з неявним зворотнім зв'язком (implicit feedback).

Так, наприклад, рекомендації за оцінками з п'ятибальною шкали - приклад завдання з явним зворотнім зв'язком. Рекомендаційні системи, що керуються фактами покупок, відвідуванням сторінок - приклади завдань з неявним зворотнім зв'язком.

У разі неявного зворотного зв'язку є невизначеність в тому, позитивно чи негативно впливають конкретний акт зворотного зв'язку на ступінь вподобання. Купівля товару в інтернет-магазині може означати досягнення користувачем своєї споживчої мети (позитивну перевагу), але в той же час покупець міг після отримання товару в ньому розчаруватися і правильно було б зарахувати негативну ступінь переваги. Очевидно, що відвідування сторінок користувачами веб-сервісу можуть відбуватися при абсолютно різному ступені зацікавленості користувача в контенті.

Варто відзначити досить типову ситуацію, коли рекомендаційній системі подаються на вхід виключно позитивні приклади взаємодії користувачів і об'єктів. Наприклад, веб-сервіс Twitter не має функціональності, що дозволяє користувачеві висловити своє низьке вподобання до контенту, а наявна тільки можливість показати свою позитивну реакцію на той чи інший контент, поширивши його своїм підписникам за допомогою функції “репост”. Подібний зворотній зв'язок

користувача дуже надійно (в порівнянні з іншими, наприклад з вподобанням) вказує на позитивну реакцію користувача на контент.

1.4 Врахування контексту в рекомендаційних системах

Часто для побудови релевантного списку рекомендацій, для рекомендаційної системи потрібна інформація про контекст, в якому веб-сервіс генерує рекомендацію для користувача. Ось деякі приклади такої контекстної інформації:

- географічне розташування мандрівника (наприклад, в разі рекомендації пам'яток;
- стан списку збережених відеозаписів чи сторінок акторів;
- сезон, час;
- погода та інші фактори.

У простих випадках (як, наприклад, з географічним розташуванням), досить фільтрувати об'єкти рекомендацій деяким правилом. У більш складних випадках (наприклад, погода) важко побудувати однозначно раціональні правила фільтрації об'єктів.

1.5 Висновки до розділу

В даному розділі було розглянуто структуру та основні характеристики рекомендаційної системи. Було досліджено існуючі підходи до побудови відеорекомендаційної системи, їх переваги та недоліки. Проаналізовано особливості врахування вхідних даних рекомендаційними системами та розглянуто показники оцінки ефективності рекомендаційних систем.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ ПІДХОДІВ ДО ПОБУДОВИ РЕКОМЕНДАЦІЙНОЇ СИСТЕМИ ТА ОПИС МЕТОДУ ПРОГНОЗУВАННЯ ОЦІНОК КОРИСТУВАЧІВ

2.1 Вступ

Для підвищення цільових показників ефективності роботи будь-якої веб платформи, в тому числі відео платформи, впроваджуються рекомендаційні системи, що створюють персоналізовані пропозиції контенту для користувачів. Для побудови таких систем використовуються моделі, що прогнозують ступінь вподобання(оцінку) користувачем нового об'єкту на основі доступних даних: це може бути, наприклад, лише історія взаємодії користувачів з платформою, або ж можуть враховуватись також екзогенні фактори, такі як опис фільму, жанр, режисер.

У цьому розділі буде проведений аналіз переваг та недоліків основних підходів, що застосовуються для побудови рекомендаційних систем.

Основною складністю при використанні відеорекомендаційних систем є генерація рекомендацій для нових користувачів, адже на момент прогнозування релевантних об'єктів система не володіє інформацією про їхні вподобання. Моделі, які на сьогодні широко застосовуються в різних галузях в рекомендаційних системах, не справляються з проблемою холодного старту, або ж вимагають перетренування моделі перед безпосередньо прогнозуванням на нових даних.

2.2 Основні підходи до побудови рекомендаційних систем

Можна виділити три основні підходи до побудови рекомендаційних систем:

- на основі ознакових описів(content-based);
- колаборативна фільтрація(collaborative filtering);
- гібридний підхід[2].

Розглянемо підхід, що ґрунтується на базі ознакових описів. Цей метод передбачає, що про користувачів і про рекомендовані об'єкти відомо досить багато інформації. Наприклад, всі користувачі заповнюють анкету, в якій вказують свою соціально демографічну інформацію, інтереси, і т.д. Про фільми з веб ресурсу з може бути відомим їх опис, основні актори, жанр, бюджет, режисер, відгуки осіб, що переглянули, інші доступні нумеричні чи категоріальні характеристики. Неструктуровані ознаки агреговано представляються у вигляді векторів (модель Vector-Space)[3]. Кожен елемент вектора - ознака, що потенційно описує зацікавленість користувача. Аналогічно, фільм чи продукт - вектор в цьому ж просторі.

З процесом взаємодії користувача та системи(наприклад, користувач переглянув фільм), векторний опис переглянутих ним фільмів об'єднується (сумується та нормалізується) в єдиний вектор і таким чином формується вектор його інтересів. Далі достатньо знайти фільм, описова характеристика якого найближча до вектору інтересів користувача, тобто вирішити задачу пошуку К найближчих сусідів.

Не всі елементи елементи є однаково значимі, наприклад, в текстовому описі сполучники не несуть жодної корисної інформації. Тому при визначенні числа співпадаючих елементів в двох векторах всі виміри потрібно попередньо зважувати по їх значущості. Для цього використовується перетворення TF-IDF, яке надає більшу вагу більш рідким інтересам, співпадіння таких інтересів має більше значення при визначенні близькості двох векторів, ніж співпадіння популярних. Принцип TF-IDF тут застосовується і до звичайних номінальних атрибутів, як то жанр, режисер, мова. TF(Term Frequency) - міра значимості атрибуту для користувача, IDF - міра рідкості атрибуту серед всіх користувачів.

В якості міри для визначення близькості векторних представлень фільмів найчастіше використовується косинусна відстань:

$$sim(A, B) = cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

Якщо користувач додав нову оцінку, вектор його інтересів оновлюється інкрементально, тобто перераховуються лише ті елементи, що змінились. Також ефективним є підхід, за яким новим оцінкам надається більша вага, оскільки вподобання користувачів можуть змінюватись з плином часу.

На практиці, використання такого підходу сильно обмежене, оскільки збір описової інформації про користувачів та об'єкти дуже дорога процедура, яку часто неможливо організувати не на шкоду якості використання сервісу, що робить рекомендаційну систему невиправдано дорогою.

Коллаборативна фільтрація. Колаборативною фільтрацією називається прогнозування ступеня вподобання за умови, коли рекомендаційна система

не володіє будь-якою описовою інформацією про користувачів і об'єкти (або не використовує), будує прогноз виключно на основі взаємодії користувачів з об'єктами, доступні дані фактично описуються матрицею, елементи якої - оцінки кожним користувачем кожного фільму, якщо такі були (рисунок 2.1). Ідея цього підходу полягає в тому, щоб якщо користувач здійснював перегляд контенту, знайти користувачів зі схожими смаками, і порекомендувати даному користувачеві те, що схожі на нього люди переглядали, а користувач ще ні.

Рисунок 2.1– Матриця вподобань об'єктів

Нехай:

U - множина користувачів (users),

I - множина об'єктів (items).

Інформація про відомі вподобання представлена у вигляді набору трійок:

$$D = \{(u, i, r_{ui}) \mid (u, i) \in R\},$$

де $r_{ui} \in R$ — ступінь вподобання об'єкта $i \in I$ користувачем $u \in U$;

$R \subseteq U \times I$ — множина пар (користувач, об'єкт), про які відомий ступінь вподобання;

$R(u) = \{i : (u, i) \in R\}$ - множина об'єктів, суміжних з користувачем u ;

$R(i) = \{u : (u, i) \in R\}$ - множина об'єктів, суміжних з об'єктом i ;

По відомій інформації D потрібно вміти будувати прогноз вподобання $\hat{r}_{ui} \approx r_{ui}$ для нових пар $(u, i) \notin R$.

Матриця оцінок - $R \in (R \cup \emptyset)^{|U| \times |I|}$, рядки якої відповідають користувачам, стовпці — об'єктам, а елементи приймають значення r_{ui} , якщо $(u, i) \in R$, інакше — пропуск \emptyset .

Висувається три типи припущень про матрицю оцінок R , що використовується в алгоритмах колаборативної фільтрації

- всі пропуски в матриці R відбулися випадково (MAR, missing-at-random);
- всі пропуски є наслідком негативного вподобання (AMAN, all-missing-are-irrelevant);
- пропуски в матриці R відбулися не випадково (MNAR, missing-not-at-random).

Під «випадковістю» вище мається на увазі ймовірнісне припущення про зміщеність розподілу пропущених оцінок в негативний бік.

Алгоритми колаборативної фільтрації, що використовують гіпотези AMAN і MNAR історично прийнято називати «застосовними до даних з неявним зворотнім зв'язком».

Ідея урахування гіпотез AMAN / MNAR породила групу алгоритмів колаборативної фільтрації, у яких налаштовуються латентні фактори користувачів і об'єктів не тільки на відомі елементи матриці R , але й на пропущені. Пропущені елементи, за припущенням, мають негативну оцінку вподобання, але при цьому впливають на параметри моделі з меншою вагою, ніж відомі.

Тож задачу колаборативної фільтрації можна розглядати як задачу заповнення пропущених значень у матриці.

Підходи до вирішення завдання колаборативної фільтрації умовно можна розділити на дві великі групи:

- на основі евристик (memory / heuristic-based),
- на основі побудови моделі переваги (model-based).

Розглянемо методи на основі евристик. До цієї групи методів відносяться алгоритми, що виражають пропущені значення безпосередньо через елементи матриці оцінок.

Прикладом алгоритму на основі евристик колаборативної фільтрації є зважування ступеня переваги по користувачах (user-based) і по об'єктах (item-based).

$$\hat{r}_{ui} = \bar{r}_u + \frac{1}{\sum_{u' \in R(i)} |sim(u, u')|} \sum_{u' \in R(i)} sim(u, u')(r_{u',i} - \bar{r}_{u'}), \quad \text{зважування по}$$

користувачам;

$$\hat{r}_{ui} = \bar{r}_i + \frac{1}{\sum_{i' \in R(i)} |sim(i, i')|} \sum_{i' \in R(i)} sim(i, i')(r_{u,i'} - \bar{r}_{i'}), \quad \text{зважування по об'єктах,}$$

де $\bar{r}_u = \frac{1}{|R(u)|} \sum_{i \in R(u)} r_{ui}$ - середнє значення переваги по користувачам;

$\bar{r}_i = \frac{1}{|R(i)|} \sum_{u \in R(i)} r_{ui}$ - середнє значення переваги по об'єктам;

$sim(u, u')$ - наперед задана метрика схожості користувачів;

$sim(i, i')$ - наперед задана метрика схожості об'єктів.

Міра схожості $sim(u, u')$ (і аналогічна для об'єктів) обчислюється по матриці оцінок R , або з використанням додаткової інформації про користувачів (об'єктів). Міра схожості є важливим параметром алгоритму.

Найбільш використовувані прості метрики схожості - кореляція Пірсона та косинусна відстань між відповідними рядками (стовпцями) матриці оцінок:

$$sim(u, u') = \frac{\sum_{i \in R(u) \cap R(u')} (r_{u,i} - \bar{r}_u)(r_{u',i} - \bar{r}_{u'})}{\sqrt{\sum_{i \in R(u) \cap R(u')} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in R(u) \cap R(u')} (r_{u',i} - \bar{r}_{u'})^2}} - \text{кореляція Пірсона};$$

$$sim(u, u') = \frac{\sum_{i \in R(u) \cap R(u')} r_{u,i} r_{u',i}}{\sqrt{\sum_{i \in R(u)} r_{u,i}^2} \sqrt{\sum_{i \in R(u')} r_{u',i}^2}} - \text{косинусна відстань.}$$

Кореляція Пірсона розраховує лінійну залежність між двома об'єктами (користувачами) як функцію їх параметрів. Але цей алгоритм не враховує міру близькості по всьому набору об'єктів. Вибірку потрібно заздалегідь обробити, відфільтрувавши по спаданню показника близькості (високорівневий показник схожості).

Подібні алгоритми практичні для одноразового обчислення рекомендацій на розподіленому кластері, добре підходять для обчислювальної архітектури MapReduce, проте погано підходять для швидкого оновлення рекомендацій. Налаштування міри схожості для цієї задачі є однією з ключових проблем, адже навіть для конкретної галузі немає налагодженого алгоритму.

Іншим вагомим недоліком алгоритмів на основі евристик є неадекватність прогнозування в умовах сильної розрідженості матриці оцінок R (в сенсі пропущених значень), що призводить до неможливості обчислення метрик схожості користувачів чи об'єктів (в випадку, коли множина $R(u) \cap R(u')$ пуста). Сильна розрідженість матриці оцінок може бути наслідком проблеми холодного старту, проте в деяких галузях вона завжди сильно розріджена і не може стати достатньо щільною.

Алгоритми на основі побудови моделі переваги. В цих підходах потрібно обрати функцію $r^*(u,i;\theta)$ з деякого параметризованого сімейства моделей $\theta \in \Theta$. Вибір відбувається, наприклад, шляхом мінімізації регуляризованого емпіричного ризику

$$L(\theta) = (r^{\wedge}(u, i; \theta), r_{ui}) + \lambda \Omega(\theta) \rightarrow \min, \theta \in \Theta, (u, i) \in R,$$

де $l(\hat{r}, r)$ — функція втрат регресії,

λ — сила регуляризації,

$\Omega(\theta)$ — регуляризатор на множині параметрів Θ .

У систем колаборативної фільтрації є кілька загальних проблем, які повинна вирішувати будь-яка модель. Матриця оцінок, як правило, розріджена - зазвичай і користувачів, і об'єктів багато, а оцінок набагато менше, адже середній користувач оцінює зовсім небагато об'єктів; інші ж елементи матриці нам невідомі, і саме їх і треба спрогнозувати. Також для систем в основі яких лежить колаборативна фільтрація, а також для систем на основі ознакових описів об'єктів виникає проблема холодного старту. Для користувачів - коли приходить новий користувач, у якого ще немає оцінок, незрозуміло, як сформувати для нього релевантний список рекомендацій. Якщо зовсім немає оцінок - можна рекомендувати найпопулярніші об'єкти, але це вимагає мануальної обробки в системі таких випадків, що може призвести до нестабільності й погіршення якості роботи сервісу. Проблема холодного старту є і для об'єктів(у випадку систем з колаборативною фільтрацією): скільки потрібно оцінок для нового об'єкту, перш ніж його можна буде впевнено рекомендувати[4].

Гібридний підхід. Незважаючи на те, що алгоритми колаборативної фільтрації на практиці показують високі показники ефективності, використання додаткової інформації може зробити показники ще вищі. Одним з недоліків колаборативної фільтрації в порівнянні з методами, заснованими на просторі ознак описі, є проблема холодного старту .

Гібридний підхід використовує композиції алгоритмів заснованих на ознакових описах і результатів колаборативної фільтрації.

2.3 Опис GraphSAGE для індуктивного представлення даних

GraphSAGE - це ітеративний алгоритм, що навчається графовим ембедінгам(векторним представленням) для кожної вершини в заданому графі. Основною перевагою цього методу є можливість отримання індуктивних ембедінгів вершин графу, як у випадку навчання на нерозмічених даних. Тобто це означає, що в разі додавання нових вершин не потрібно перетреновувати модель. GNN(Графові нейронні мережі) можуть використовувати ембедінги для великої кількості різних задач, включаючи класифікацію вершин, прогнозування взаємодії, визначення схожих об'єктів, аналізу мереж та для інших.

До створення GraphSAGE, більшість моделей, що базуються на ембедінгах вершин використовували алгоритми спектрального розкладу або матричної факторизації. Основним недоліком таких методів є те що вони за своєю суттю є трансдуктивними, тобто не вміють працювати з даними, яких не бачили раніше. Під час прогнозування ці методи очікують, що структура графу буде такою ж, як і під час навчання моделі. Якщо нова вершина буде додана до графу після створення моделі, потрібно буде знову перетреновувати модель.

На противагу, індуктивний підхід є єдиним, що забезпечує генералізацію на нових даних, що суттєво покращує ефективність рекомендаційної системи під час прогнозу.

Основна ідея в основі алгоритму GraphSAGE - те, що будь-яку вершину(в контексті відеорекомендаційних систем вершинами виступають об'єкти рекомендацій та користувачі) можна представити, використовуючи інформацію про вершини, з якими вона сполучена ребрами(оцінка користувачем фільму).

Завдання GraphSAGE вивчити репрезентацію h кожної вершини графу на основі деякої комбінації його сусідніх вершин.

Кожна вершина може мати власний векторний опис, що задається вхідними даними. Припустимо, що всі ознакові вектори для кожної вершини мають однакову розмірність. Один етап GraphSAGE може запускатись для k ітерацій. Таким чином, існує векторне представлення для кожної вершини на кожній із ітерацій.

Введемо наступні позначення:

X_v - ознакові характеристики для випадкової вершини v ;

h_0^v - вихідне ембеддингове представлення вершини v ;

h_k^v - ембеддингове представлення вершини v на ітерації k ;

z_k - ембеддингове представлення вершини v після завершення виконання GraphSAGE.

Оскільки кожна вершина може бути визначена за допомогою своїх сусідів, ембеддинг для вершини A може бути отриманий комбінацією ембеддингів сусідніх вершин. Після завершення одного етапу алгоритму GraphSAGE ми отримаємо нове представлення для вершини A (рисунок 2.2). Цей процес виконується для усіх вершин в вихідному графі.

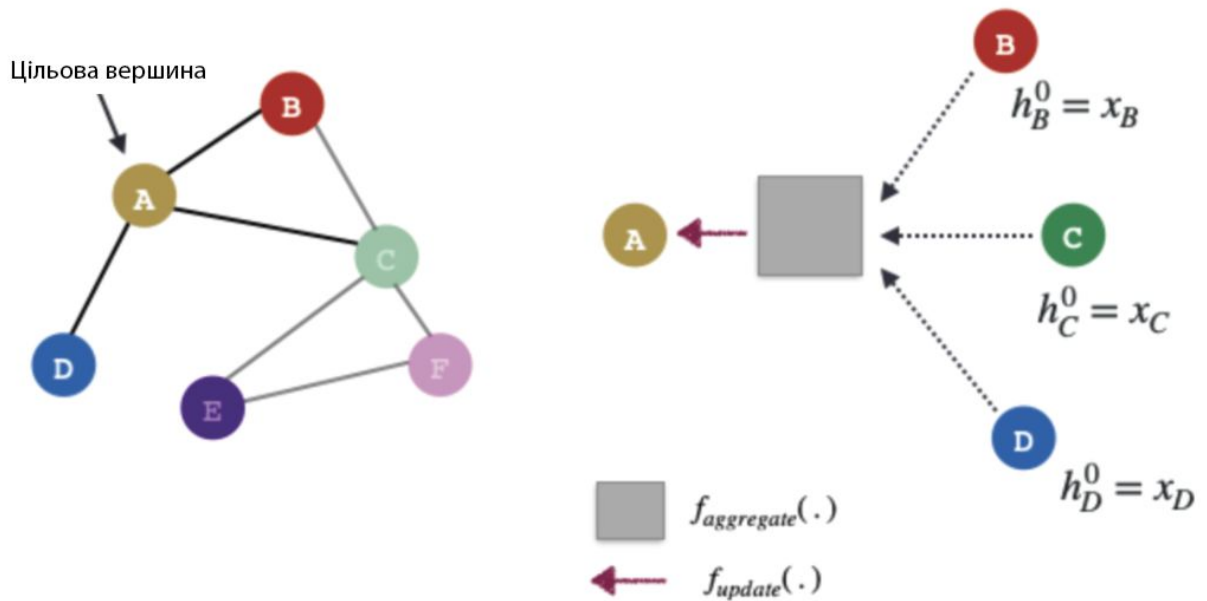


Рисунок 2.2 – Структурний опис етапу виконання GraphSAGE

GraphSAGE алгоритм складається з двох основних етапів. Оскільки це ітеративний алгоритм, спочатку проводиться ініціалізація, яка встановлює у відповідність ембедингу кожної вершини її векторний ознаковий опис ($h_v^0 = x_v$). Основні етапи алгоритму:

- Агрегація

На цьому кроці відбувається агрегація репрезентацій сусідніх вершин для цільової вершини. Функцією для агрегації може виступати будь-яка диференційовна функція. Це може бути просте усереднення або ж нейронна мережа.

$$a_v = f_{aggregate}(\{h_u | u \in N(v)\}),$$

де a_v - агреговане представлення вершини v .

- Оновлення.

Після отримання агрегованого представлення вершини v на основі її сусідніх вершин оновлюємо поточну вершину v використовуючи комбінацію її попередніх представлень і агрегованого представлення. Функцією оновлення може виступати диференційовна функція, наприклад усереднення чи більш складна, як-то нейронна мережа:

$$h_k^v = f_{update}(a_v, h_{k-1}).$$

GraphSAGE ґрунтується на агрегації та оновленні репрезентацій вершин. Параметр k визначається кількістю сусідніх вершин та етапів для обчислення представлення вершини v .

Наприклад, замість ініціалізації представлення вершини B , можна використати результати агрегації та оновлення для вершини B на основі її сусідніх вершин. Це ж можна провести і для вершин C та D в етапі $k = 1$ (рисунок 2.3). На етапі $k = 0$ ми повинні ініціалізувати ембеддинги сусідніх вершин вихідними ознаковими описами[5].

На кожній ітерації $k \in \{1, \dots, K\}$ для кожної вершини $\{v \in V\}$ обчислюємо агрегацію та оновлюємо вектор представлень:

$$h_{N(v)}^k = f_{\text{aggregate}}(\{h_u^{k-1}, \forall u \in N(v)\});$$

$$h_v^{k'} = \sigma(W^k, \text{concat}(h_v^{k-1}, h_{N(v)}^k)),$$

де $\sigma(x) = \frac{1}{1 + e^{-x}}$.

В кінці кожної ітерації нормалізуємо ембеддінги вершин:

$$h_v^k = \frac{h_v^{k'}}{\|h_v^{k'}\|_2}, \forall v \in V.$$

В якості функції агрегації може усереднення або LSTM агрегація. Однак LSTM агрегація вимагає перемішування вхідних міток. Матриця вагових коефіцієнтів є окремою для кожної ітерації, тобто в процесі тренування модель вивчає важливість конкретних сусідів для цільової вершини. Для запобігання проблеми збурення градієнтів використовується нормалізація ембеддінгів вершин.

Розглянемо, як відбувається тренування графової нейронної мережі GraphSAGE. Модель GraphSAGE може використовуватись як для задачі навчання по розмічених даних, так і для задачі навчання на нерозмічених даних. Навчання на розмічених даних зводиться до задачі класифікації вершин за допомогою обчислення крос-ентропії під час тренування моделі. У

випадку задачі на нерозмічених даних алгоритм намагається зберегти структуру графу за допомогою мінімізації функції втрат (рисунок 2.4).

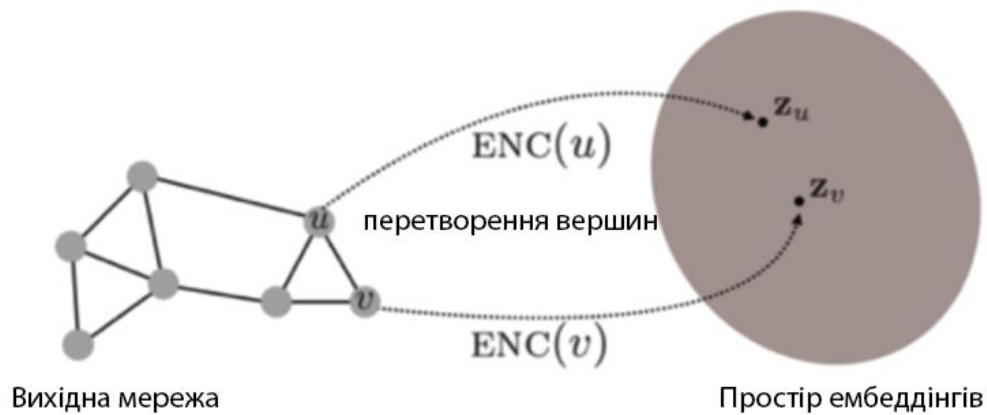


Рисунок 2.4 – Схема кодування вершин

$$l(z_v) = -\log(\delta(z_u^T z_v) + \varepsilon) - Q * E_{v_n \sim P_n(v)} \log(\delta(-z_u^T z_v) + \varepsilon)$$

Перша складова функції втрат намагається домогтися того, що якщо вершини u та v є близькими в поточному графі, їх векторне представлення вершин повинне бути семантично близьким. В ідеальному випадку очікується, що значення z_u та z_v є великими. Значення сигмоїди в результаті повинне бути близьким до одиниці, а логарифм відповідно до нуля.

Другий елемент функції втрат намагається домогтися протилежного. Тобто, якщо вершини u та v є далекими в поточному графі, ми очікуємо що їх векторні представлення будуть протилежними/далекими. В ідеальному випадку ми очікуємо, що добуток z_u та z_v буде значно меншим нуля. В

результаті отримаємо додатне число під функцією сигмоїди. Вона набуватиме значення 1, і логарифм відповідно - нуля. Оскільки потенційно може бути велика кількість далеких від цільової вершин, ми беремо до уваги тільки кілька далеких вершин $P_n(v)$. Таким чином отримуємо збалансовану функцію втрат під час тренування[6].

2.4 Модифікація GraphSAGE для врахування гетерогенних даних

Алгоритм GraphSAGE може бути узагальнений для гетерогенних графів, тобто для графів, що мають вершини та ребра різних типів.

Щоб підтримувати роботу з гетерогенними вершинами та ребрами можна розширити GraphSAGE алгоритм з введенням окремої матриці W_{neigh} вагових коефіцієнтів для кожної унікальної впорядкованої пари (n_1, e, n_2) , $n_1 \in N_1$, $n_2 \in N_2$ - типи вершин. В додаток, для підтримки гетерогенності моделлю введемо окремі матриці W_{self} для вершин кожного типу.

Розглянемо етапи оновлення для агрегації усереднення та GCN агрегації:

- HinSAGE для агрегації усередненням.

Агрегація даних сусідніх вершин:

$$h_{N_r(v)}^k = \frac{1}{|N_r(v)|} \sum_{u \in N_r(v)} D_p[h_u^{k-1}].$$

Оновлення може відбуватись кількома способами: конкатинацією та сумуванням. Оновлення конкатинацією:

$$h_v^k = \delta(\text{concat}[W_{t_v, self}^k D_p[h_v^{k-1}], W_{1, neigh}^k h_{N_1(v)}^k, \dots, W_{R_e, neigh}^k h_{N_{R_e}(v)}^k] + b^k)$$

оновлення сумуванням:

$$h_v^k = \delta(W_{t_v, self}^k D_p[h_v^{k-1}] + W_{r_e, neigh}^k h_{N_{r(v)}}^k) + b^k.$$

– HinSAGE для GCN агрегації.

Розглянемо цикл агрегації та оновлення. Агрегація:

$$h_{N_r(v)}^k = \frac{1}{|N_r(v)|+1} (h_v^{k-1} + \sum_{u \in N_r(v)} h_u^{k-1}).$$

Оновлення на шарі k:

$$h_v^k = \delta(\frac{1}{R_e} W_r^k \cdot h_{N_{r(v)}}^k + b^k),$$

де W_r^k - тренувальні матриці вагових коефіцієнтів $[d_k, d_{k-1}]$, (одна на ребро типу r).

Кількість параметрів, що тренуються на кожному шарі k для цієї моделі: $R_e d_k d_{k-1} + d_k$, тобто GCN агрегація в випадках малих датасетів менш схильна до перенавчання[7].

2.5 Висновки до розділу

В цьому розділі було проаналізовано основні переваги та недоліки існуючих підходів до побудови рекомендаційних систем; розглянуто модель GraphSAGE для роботи з графовим представленням вхідних даних, а також можливість його розширення для підтримки роботи з гетерогенними вхідними даними.

РОЗДІЛ 3. АНАЛІЗ ВХІДНИХ ДАНИХ, ПОБУДОВА МОДЕЛІ ДЛЯ ПРОГНОЗУВАННЯ ОЦІНОК КОРИСТУВАЧІВ ТА АРХІТЕКТУРА ВІДЕОРЕКОМЕНДАЦІЙНОЇ СИСТЕМИ

3.1 Вступ

У цьому розділі буде проаналізовано вхідних набір доступних даних, побудовано модель з урахуванням особливостей даних та проведено аналіз результатів прогнозування оцінок користувачів отриманої моделі.

За основу для досліджень було взято набір даних з бази даних Movielens. Movielens - набір даних, зібраний дослідницькою групою GroupLens. База даних містить датасети за різний історичний період часу, в залежності від розміру датасету. Найменший з доступних містить 100000 оцінок від 1000 користувачів про взаємодію з 1700 відео матеріалами на платформі. Також доступна інформація про користувачів та фільми.

Попередня обробка, аналіз історичних даних, побудова моделі для рекомендаційної системи та аналіз результатів було виконано за допомогою мови програмування Python у веб-орієнтованому інтерактивному середовищі Jupyter Notebook.

3.2 Вибір платформи та мови програмування

Для проведення дослідження та створення відеорекомендаційної системи було використано мову програмування Python. Вибір обумовлено

тим, що вона підтримується на операційних системах, широко застосовується для аналізу даних, що зумовлює існування значної кількості допоміжного навчального матеріалу та фреймворків, у яких реалізовані основні підходи та алгоритми даної галузі. Мова програмування R також дозволяє проводити аналіз великих об'ємів даних та містить реалізації основних статистичних методів. Основною перевагою мови Python перед R при її виборі була можливість її легкого використання для виведення інформації на веб-сайті чи додатку. При побудові моделі та дослідженні даних використовувались додаткові бібліотеки для Python. Для підтримки та функціонування системи їх використання є не обов'язковим, оскільки достатньо буде оперувати вже створеною моделлю.

Використані бібліотеки:

- pandas – для роботи з таблицями;
- seaborn, matplotlib, plotly – для візуалізації даних;
- numpy – для роботи з масивами даних;
- statsmodels – для моделювання;
- keras, tensorflow - для побудови нейронної мережі;
- stellagraph - для роботи з графами;
- scipy – для аналізу даних;
- multiprocessing - для паралелізації обчислень

Структура програмного продукту складається з кількох файлів jupyter notebook з реалізацією моделей прогнозування оцінок користувачів та

додаткових вище зазначених бібліотеки. В окремому файлі jupyter notebook проведено аналіз тренувального набору даних.

3.3 Аналіз вхідних даних

Для проведення дослідження було взято набір даних Movielens, зібраних науково-дослідницькою групою GroupLens. Ці дані містять інформацію про взаємодію користувачів та платформи з кінофільмами movielens.org. За основу було обрано датасет, що складається зі 100000 оцінок, проставлених 1000 користувачів на 1700 фільмів. Також датасет містить набір таблиць з додатковою інформацією про користувачів платформи, наприклад: вік, стать, вид діяльності, поштовий індекс. Можна також отримати додаткову інформацію про фільми на основі вказаних назв фільмів.

Кожен користувач оцінив щонайменше 20 фільмів. Рейтинг користувачі проставляли фільмам за шкалою від 1 до 5. Середня оцінка на платформі - 3.6, медіанна - 4 (рисунок 3.1).

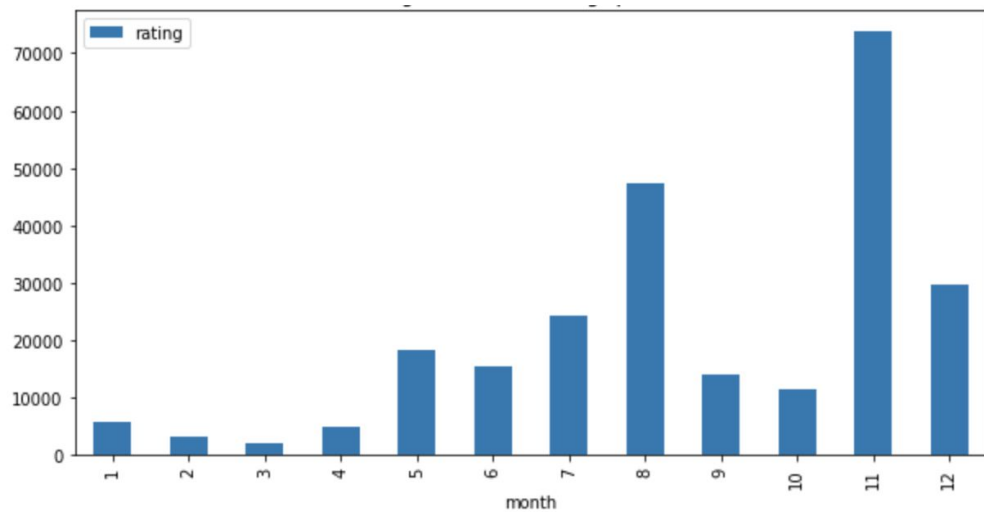


Рисунок 3.1 – Розподіл оцінок користувачів в залежності від місяця перегляду

Бачимо, що близько 70% усіх оцінок припадає на один місяць(листопад), тобто в цей період система буде найбільш навантажена.

Проаналізуємо розподіл оцінок користувачів (рисунок 3.2).

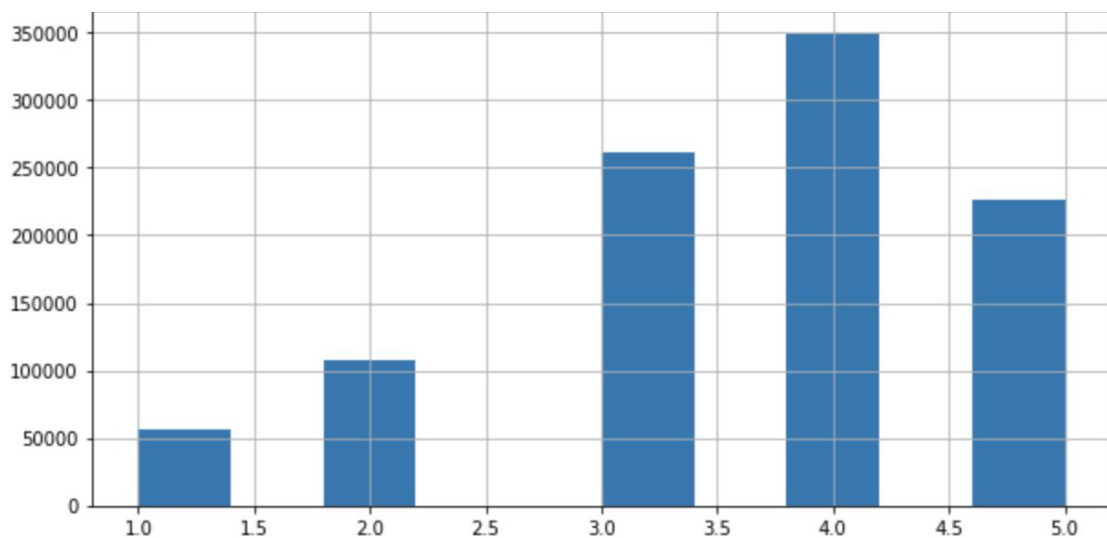


Рисунок 3.2 – Розподіл оцінок користувачів

Бачимо, що 35% оцінок становлять четвірки, 26% - трійки, 21% - п'ятірки. Це дає можливість зробити припущення про те, яким повинен бути бейзлайн, тобто базова модель прогнозування оцінок користувачів, з якою будемо порівнювати точність побудованої нами моделі. В таких задачах побудови рекомендаційних систем і прогнозування оцінок досить сильним є прогноз прогнозування середнім, навіть найсучасніші архітектури не можуть значно підвищити точність прогнозу у порівнянні з прогнозом середнім чи медіанним значенням.

Проаналізуємо оцінки фільмів в залежності від користувачів(рисунок 3.3).

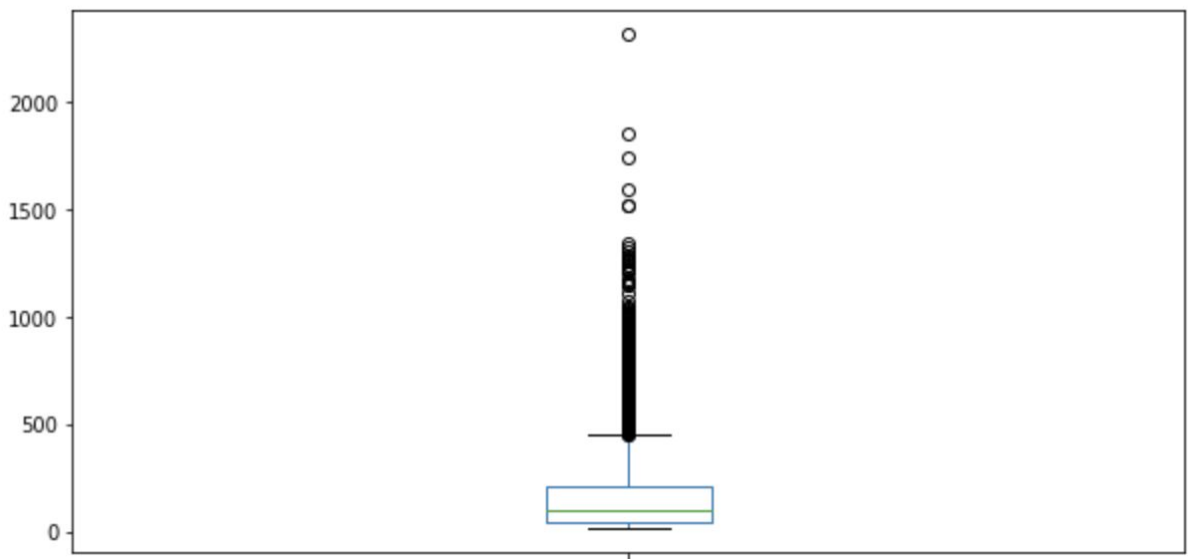


Рисунок 3.3 – Діаграма Бокса-Кокса розподілу кількості оцінених фільмів кожним користувачем

З діаграми видно, що медіанне значення кількості оцінок проставлених оцінок одним користувачем близько 100. Також важливим є те, що в даних є викиди, тобто користувачі, що оцінили неприродньо велику кількість фільмів. Такі оцінки краще відфільтрувати з тренувального набору даних, тому оцінки користувачів, що оцінили більшу за 99 перцентиль кількість фільмів я не враховувала при побудові моделі.

Проведемо аналіз доступної інформації про користувачів, адже однією з важливих переваг запропонованого мною підходу над вже існуючими(колаборативна фільтрація) є можливість врахувати додаткову інформацію про користувачів чи об'єкти.

Доступні дані про вік, стать, поштовий індекс та вид діяльності користувача. Бачимо, що в деяких користувачів інформація про вік є ймовірно несправжньою, це теж потрібно врахувати при побудові моделі (рисунок 3.4).

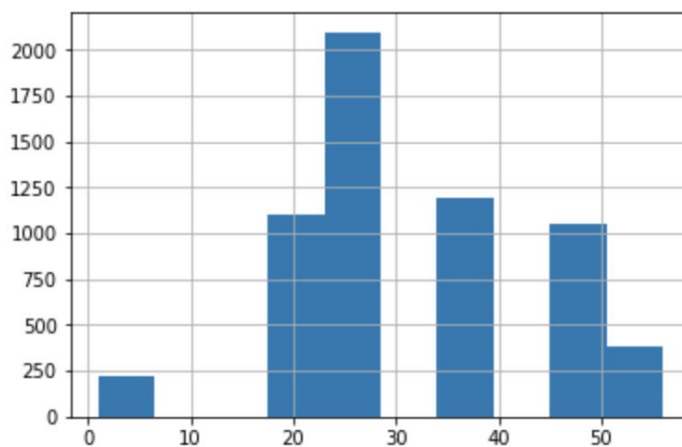


Рисунок 3.4 – Розподіл користувачів по вікових групах

Також розглянемо доступну інформацію про фільми. Доступні для поля містять дані про жанр, назву та рік зйомки. Інформацію про жанр я подавала в модель, попередньо обробивши за допомогою підходу One-Hot-Encoding. Рік зйомки передавався в модель, тобто в ознаковий опис вершин графу, що відповідали за фільми, як нормалізоване нумеричне значення. Назви фільмів також можна подати в модель у вигляді ембедингів, отриманих застосуванням TF-IDF підходу, або ж нейронної мережі word2vec чи glove. Однак додавання інформації про назву фільму не підвищило точність прогнозу, на відміну від жанру (рисунок 3.5 - 3.6).

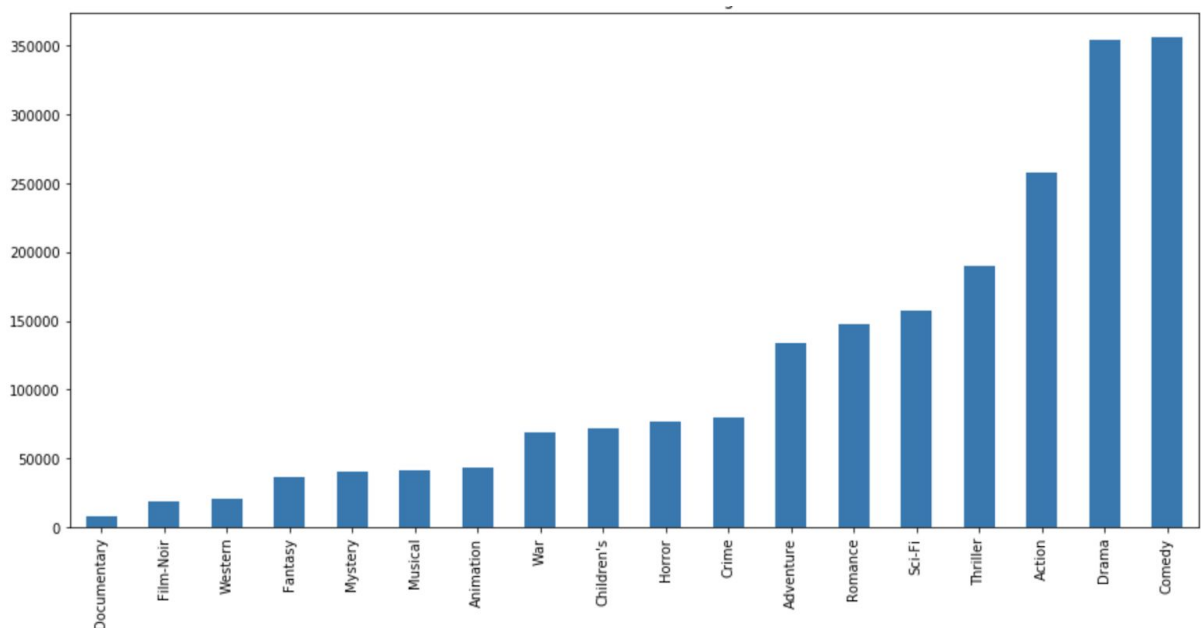


Рисунок 3.5 – Розподіл кількості оцінених фільмів за жанрами

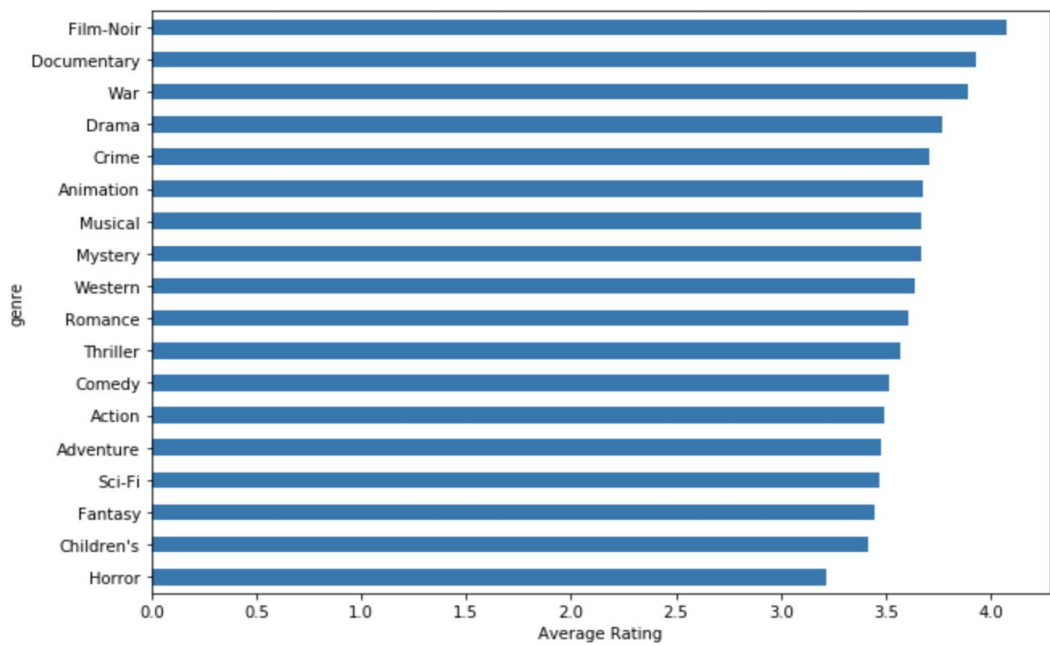


Рисунок 3.6 – Розподіл оцінок в залежності від жанру фільму

Також в ході аналізу було визначено, що немає суттєвої відмінності між розподілом оцінок фільмів різними віковими групами, тому інформацію про вік в модель я не додавала.

Провівши аналіз динаміки оцінок по жанрах, бачимо що з часом уподобання користувачів можуть змінюватись (рисунок 3.7). Це означає, що додавання інформації про період, на який прогнозуємо оцінку, в модель може підняти її точність. Однак існуючі підходи не дозволяють зробити це безпосередньо, тому єдиним виходом в перспективі є надання різних вагових коефіцієнтів спостереженням в залежності від їх актуальності.

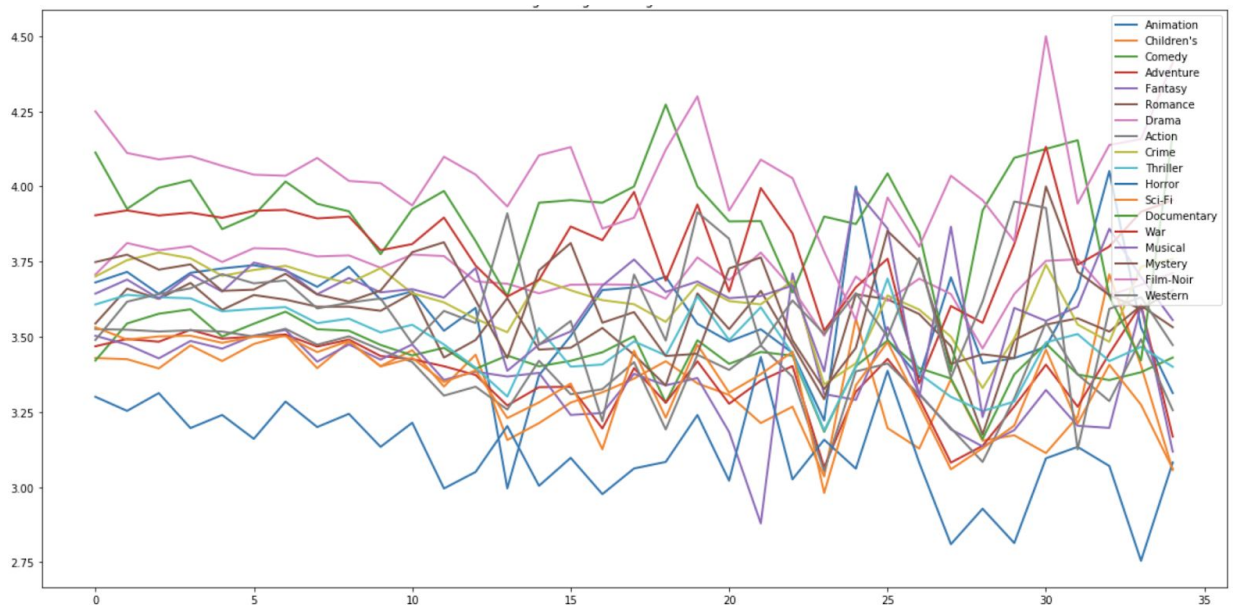


Рисунок 3.7 – Динаміка середніх оцінок фільмів по жанрах

3.4 Побудова моделі

Важливим етапом при побудові моделі є вибір валідаційної стратегії для оцінки роботи моделей. Для дослідження я розділила вибірку на тренувальну та тестову (80% та 20% від усіх спостережень відповідно). На відкладеній тестовій вибірці оцінюватимемо фінальну точність моделі (дані, яких модель не бачила під час тренування). А тренувальну вибірку розділила на власне тренувальну та валідаційну. Валідаційна вибірка необхідна для того, щоб контролювати що не відбувається перенавчання моделі на тестовій, тобто модель вміє генералізувати. Перенавчання - це явище, що описує випадок, коли помилка на тестовій вибірці значно більша ніж на тренувальній. Це головна проблема усіх підходів в машинному навчанні, адже якби не було такого ефекту (тобто помилка на тестовому наборі даних

приблизно співпадала з помилкою на тренувальному), то весь процес навчання моделі би зводився до мінімізації помилки саме на тестовій вибірці(так званий емпіричний ризик)[8].

Для побудови моделі прогнозування оцінок користувачів на датасеті MovieLens я використала узагальнення алгоритму GraphSAGE для гетерогенних графів (HinSAGE). Опис алгоритму наведено в розділі 2. Фактично ми маємо справу з задачею навчання на розмічених даних на мережі користувач-фільм з вузлами двох типів(користувачі та фільми, обидва містять свої атрибути) і зв'язками, що відповідають поставленим оцінкам у парі користувач-фільм з цілим значенням атрибутів від 1 до 5, тобто якщо користувач не проставив рейтинг фільму, в мережі відповідні вузли не є зв'язаними.

Для цього була побудована модель з наступною архітектурою: HinSAGE з двома шарами, що на вхід приймає розмічені впорядковані пари (користувач-фільм), які задають вершини та відповідну поставлену оцінку, а на виході отримуємо пару ембеддінгу вершини користувача та ембеддінгу вершини фільму. Після цього згенеровані ембеддінги подаються на вхід в регресійний шар нейромережі, щоб отримати числовий прогноз оцінки у парі фільм-користувач. Загалом модель тренується мінімізацією функції помилки. Функцією для мінімізації я обрала MSE між справжніми та прогнозованими оцінками. Мінімізація відбувається застосуванням SGD (стохастичного градієнтного спуску) для оновлення параметрів моделі по міні-батчах, сформованих з тренувальних пар користувач-фільм і відповідних їм оцінок (рисунок 3.8 - 3.10)[9].

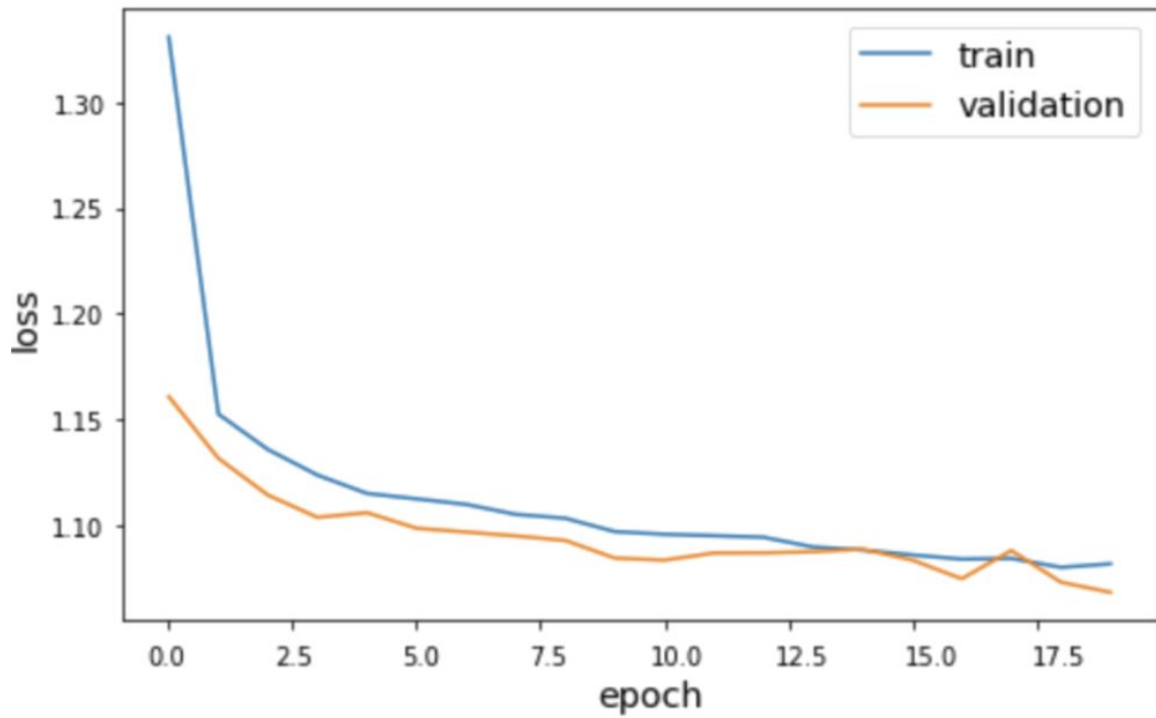


Рисунок 3.8 – Функція втрат під час навчання моделі

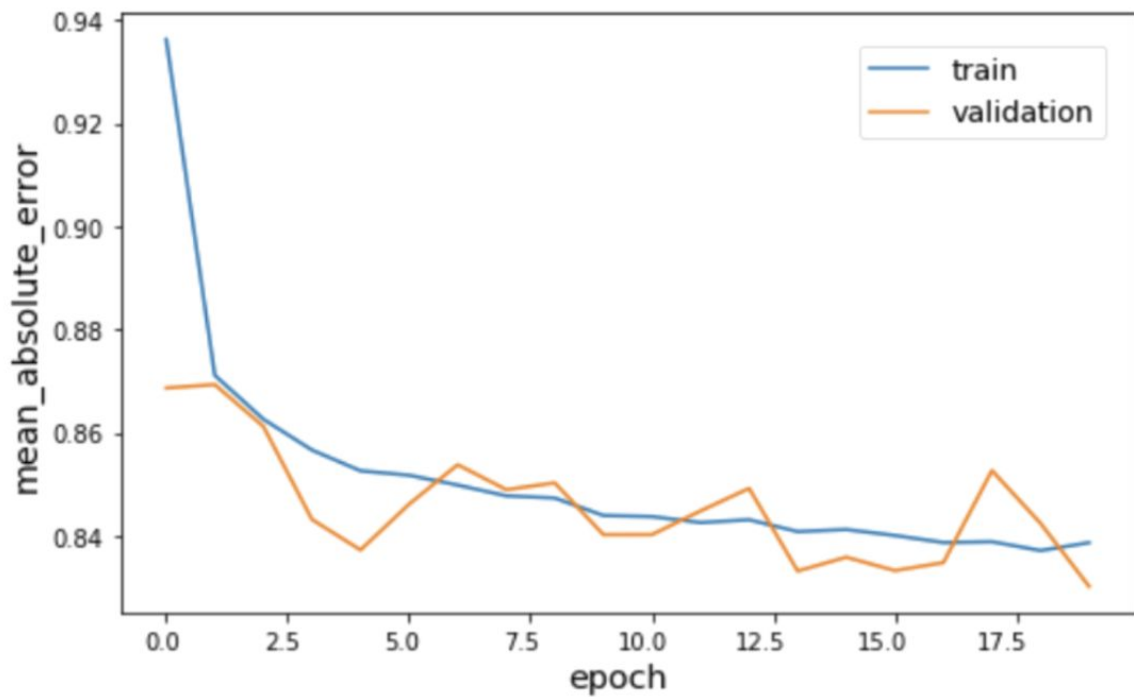


Рисунок 3.9 – Динаміка MAE під час навчання моделі

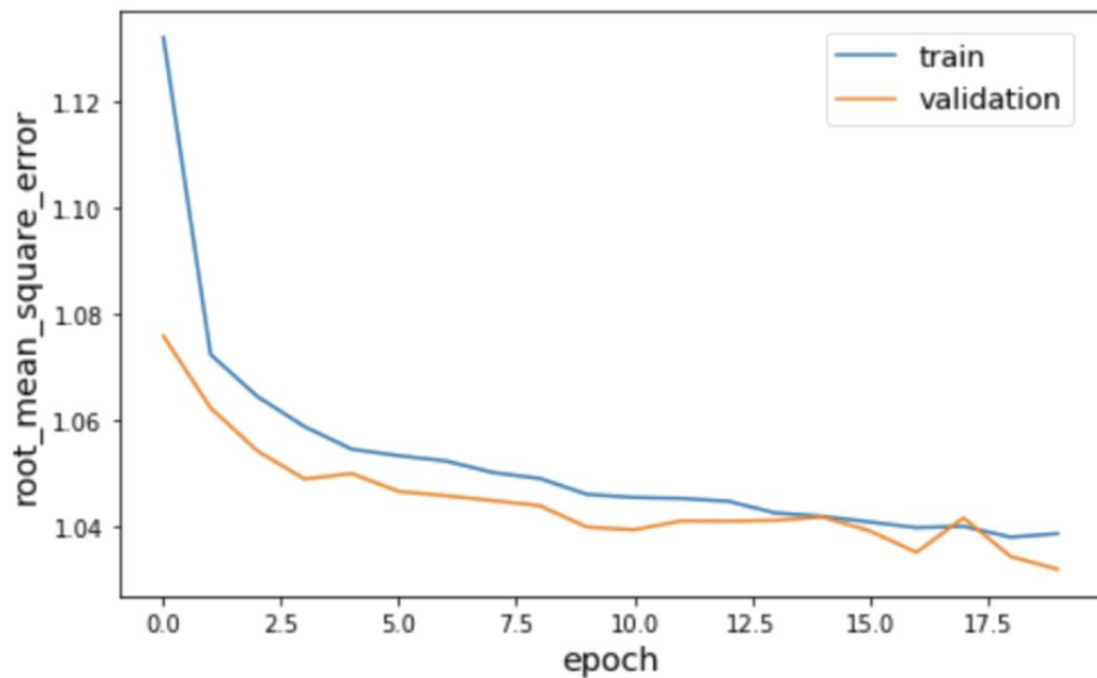


Рисунок 3.10 – Динаміка RMSE під час навчання моделі

3.5 Аналіз результатів прогнозу

Проаналізуємо точність прогнозу моделі на відкладеній тестовій вибірці. Для оцінки точності я обрала наступні метрики - RMSE та MAE.

RMSE (Root Mean Squared Error) обчислюється за наступною формулою:

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m |a_i - y_i|^2}$$

де a_i – справжні значення оцінок користувачів;

y_i – прогнозовані значення оцінок користувачів;

n – кількість спостережень [10].

Чим точнішою є модель, тим ближчим до нуля є значення похибки RMSE.

Для натренованої моделі я отримала наступні значення RMSE:

- на тренувальній вибірці 1.041;
- на валідаційній вибірці 1.034;
- на тестовій вибірці 1.035.

Оскільки значення помилки на тестовому наборі даних не сильно відрізняється від помилки на тренувальному та валідаційному наборах, можна зробити висновок, що модель навчилася генералізувати та немає проблеми перенавчання.

MAE (Mean Absolute Error) визначається за наступною формулою:

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n},$$

де y_i – прогнозовані значення оцінок користувачів;

x_i – справжні значення оцінок користувачів;

n –кількість спостережень.

Аналогічно до RMSE, чим вища точність, тим меншим є значення помилки MAE.

Для моделі я отримала наступні значення MAE:

- на тренувальній вибірці 0.842;
- на валідаційній вибірці 0.831;
- на тестовій вибірці 0.830.

Також бачимо, що модель є стабільною(це підтверджує і динаміка зміни MAE і RMSE під час навчання) і немає перенавчання.

Як було зазначено вище, в задачах створення рекомендаційних систем сильною базовою моделлю для існуючих алгоритмів є прогнозування середньої оцінки для всіх користувачів. Порівняємо модель з базовою, що прогнозує середні оцінку по фільму для кожного користувача, який ще його не бачив.

Значення RMSE для базової моделі на тестовій вибірці 1.125. Значення MAE - 0.944.

Бачимо, що по обох метриках точність моделі є значно вищою за точність базової моделі прогнозу.

3.6 Архітектура відеорекомендаційної системи

Побудована модель прогнозування оцінок користувачів є основою для побудови відеорекомендаційної системи (рисунок 3.11).

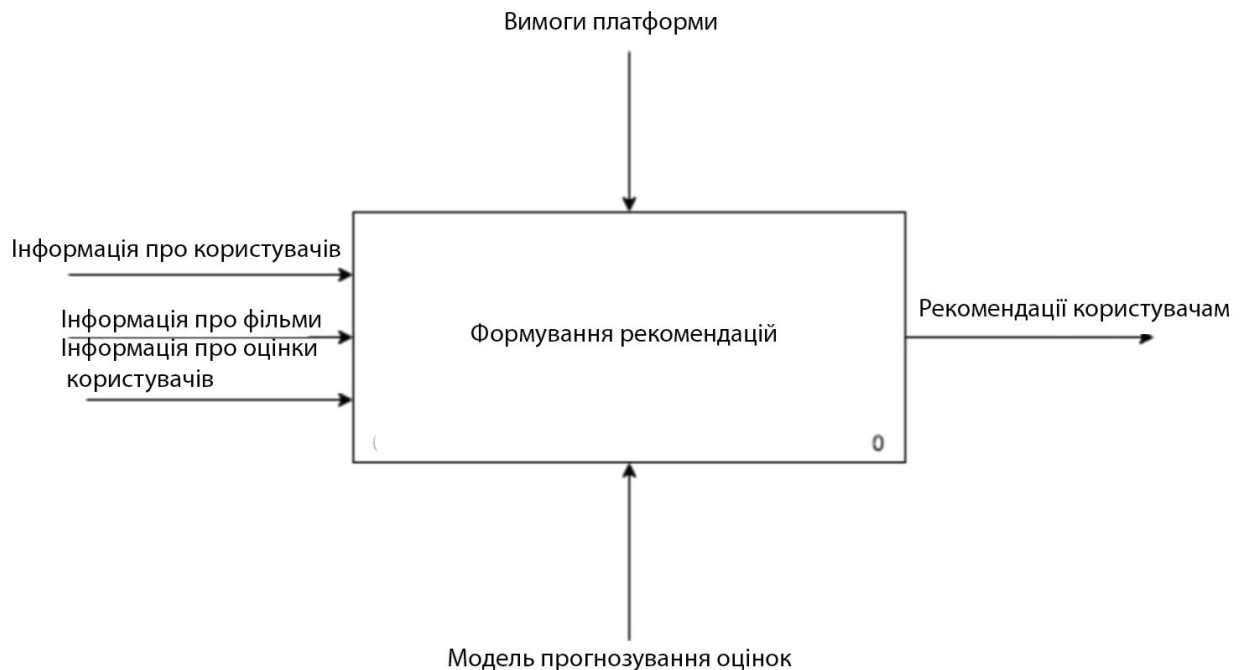


Рисунок 3.11 – Структура відеорекомендаційної системи

Для роботи рекомендаційної системи необхідні дані на вхід:

- інформація про користувача(отримується з бази даних, що запроваджена в веб платформі);
- інформація про фільм(отримується з бази даних, що запроваджена в веб платформі), а саме жанр, назва та ідентифікатор;
- історичні дані про взаємодію користувачів та відео

платформи(необхідно для побудови моделі прогнозування уподобань та регулярного її перетренування для підвищення точності)[11].

Дані про користувачів та фільми трансформуються в необхідний для прогнозу моделі формат. Для кожного з користувачів модель прогнозує оцінку ще для кожного з ще не переглянутих ним фільмів. Далі на основі прогнозів моделі список фільмів ранжується за прогнозованим рейтингом.

Об'єкти з найвищою спрогнозованою оцінкою надаються користувачеві в якості персоналізованих рекомендацій.

3.7 Висновки до розділу

У даному розділі була наведена практична частина магістерської дисертації: проведено аналіз вхідних даних, побудовано модель для прогнозування оцінок користувачів, проаналізовано результати прогнозів, описаний алгоритм для побудови відеорекомендаційної системи на основі створеної моделі.

РОЗДІЛ 4. СТАРТАП-ПРОЕКТ «Video Recommendation System»

4.1 Вступ

Ціллю даної магістерської дисертації є створення рекомендаційної системи для відеоплатформи. Існуючі підходи до створення моделей, що лежать в основі відеорекомендаційних систем є неефективними при генерації рекомендацій для нових користувачів. В цій магістерській дисертації реалізовано алгоритм, який вирішує вище зазначену проблему, а отже його використання може збільшити цільові показники ефективності платформи. На основі даного алгоритму можна запустити стартап-проект «Video Recommendation System», що в перекладі з англійської мови означає «Відеорекомендаційна система».

4.2 Опис ідеї проекту

У таблиці 4.1 надано зміст ідеї (що пропонується), можливі напрямки застосування та основні вигоди, що може отримати користувач товару.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення програмного продукту, який на основі алгоритмів, що дозволяють отримувати індуктивні представлення об'єктів, тобто генерувати персональні рекомендації для нових користувачів швидко, дозволяє створити відеорекомендаційну систему з високими цільовими показниками ефективності залучення та утримання нових користувачів	1. Точний прогноз ранжованого списку рекомендацій для нових користувачів	1. Нові користувачі платформи отримують персональні рекомендації, а не загальні, базовані на популярності контенту.
	2. Отримання ранжованого списку рекомендацій для нових користувачів без перетренування моделі	2. Суттєве підвищення швидкодії роботи платформи
	3. Автоматизація процесів	3. Дозволяє зекономити людські ресурси та позбаватись від мануальних рішень генерації рекомендацій для нових користувачів

Виділимо такі техніко-економічні характеристики ідеї:

- 1) висока точність знаходження релевантних об'єктів видачі;
- 2) зрозумілість та простота для користувача;
- 3) широка галузь для застосування та універсальність.

Для порівняння цього продукту з іншими представниками на ринку, у якості конкурентів виберемо такі три програмні продукти:

- 1) Vimeo;

2) Кинопоиск;

3) Netflix.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		VideoRecomm endation System	Кинопоиск	Vimeo	Netflix			
	Якісний програмний продукт	+	-	-	+			+
	Підтримка продукту протягом всієї експлуатації	+	-	+	+		+	
	Багатофункціональний аналіз і точний прогноз	+	+	-	-	+		

4.3 Технологічний аудит ідеї проекту

Визначення технологічної здійсненності ідеї проекту передбачає аналіз складових, наведених у таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Оренда офісу	Вибір зручного розташування	+	+
	Закупівля програмного забезпечення	Підбір програмного забезпечення, яке забезпечить всі потреби в реалізації і підтримці програми	+	+
	Закупівля технічного обладнання для реалізації	Закупівля ноутбуків, екранів та іншого обладнання для комфортної роботи всіх видів персоналу	+	+
Обрана технологія реалізації ідеї проекту: офіс з доступним програмним і технічним забезпеченням.				

4.4 Аналіз ринкових можливостей запуску стартап-проекту

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

п/п	Показники стану ринку (найменування)	Характеристика
	Кількість головних гравців, од	4
	Загальний обсяг продаж, грн	40 млн грн
	Динаміка ринку (якісна оцінка)	Зростає
	Наявність обмежень для входу	Отримання патенту та ліцензій
	Специфічні вимоги до стандартизації та сертифікації	Немає
	Середня норма рентабельності в галузі (або по ринку), %	15

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (таблиця 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Покупка реалізованої програми	Відеоплатформи	Різні масштаби (обсяг доступного контенту) відеоплатформи, доступність історичних даних	Висока точність персоналізованих рекомендацій і швидкодія рекомендаційної системи

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 4.6 і 4.7). Фактори в таблиці подані в порядку зменшення значущості.

Таблиця 4.6 – Фактори загроз

п/ п	Фактор	Зміст загрози	Можлива реакція компанії
	Погана якість даних про взаємодію користувачів з платформою	Кожна веб платформа має своє програмне забезпечення для збору даних. Якість деяких із них може бути сумнівною.	Запропонувати перейти на інше програмне забезпечення, в якості якого ми впевнені.
	Можливий людський фактор (помилки в програмній реалізації)	В кожній програмі завжди є свої помилки, що зазвичай викликає незадоволення зі сторони клієнта.	Запропонувати знижку для того, щоб клієнт не відмовлявся від підписки, або не розривав договір
	Недостатня кількість історичних даних про взаємодію користувачів та платформи та опису об'єктів рекомендацій для точного прогнозу	Низька кількість історичних даних може вплинути на якість прогнозованих даних	Попередити про це клієнта і запропонувати відкласти впровадження цієї компанії

Таблиця 4.7 – Фактори можливостей

п/п	Фактор	Зміст можливості	Можлива реакція компанії
	Додаткова можливість аналітичного супроводу	З часом можна розширювати можливості програмного забезпечення, наприклад давати аналітичний супровід отриманих результатів у вигляді графіків, висновків та порад	Залежить від бюджету і потреб власника відеоплатформи

Далі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку (таблиця 4.8).

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія / олігополія / монополістична / чиста	Чиста	Гарні перспективи розвитку
2. За рівнем конкурентної боротьби - локальний / національний / ...	Національний	Ведучи конкуренцію на національному рівні, компанії необхідно прикласти належні зусилля для охоплення всього національного ринку
3. За галузевою ознакою - міжгалузева / внутрішньогалузева	Внутрішньогалузева	Необхідно зосередити зусилля на пошуку конкурентних переваг, які дозволять компанії займати стійкі конкурентні позиції на даному ринку
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	Товарно-родова	
5. За характером конкурентних переваг - цінова / нецінова	Нецінова	Перевагою є унікальність на ринку на даний момент
6. За інтенсивністю - марочна / не марочна	Марочна	

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 4.9).

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки:	Прямих конкурентів немає	Отримання патентів і ліцензій	Немає постачальників	Немає	Немає

За результатами аналізу таблиці робиться висновок щодо принципової можливості роботи на ринку з огляду на конкурентну ситуацію (таблиця 4.10).

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
	Відносна унікальність на ринку	Існуючі на ринку підходи погано працюють при генеруванні персоналізованих рекомендацій для нових користувачів платформи
	Математично обґрунтовані підходи	Перевагою є те, що програмна реалізація є повністю математично обґрунтованою
	Повна підтримка програмного забезпечення після впровадження	Після того, як продається програмне забезпечення, воно продовжує покращуватися, виправлятися помилки і т.д.

За визначеними факторами конкурентоспроможності (таблиця 4.10) проводиться аналіз сильних та слабких сторін стартап-проекту (таблиця 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін «Video Recommendation System»

п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з «Video Recommendation System»						
			-3	-2	-1		1	2	3
	Відносна унікальність на ринку	20	+						
	Математично обґрунтовані підходи	15					+		
	Повна підтримка програмного забезпечення після впровадження	10		+					

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 4.12).

Таблиця 4.12 – SWOT- аналіз стартап-проекту

Сильні сторони: – Відсутність реальних конкурентів – Гарантія коректних результатів і математичне обґрунтування	Слабкі сторони: - Програмне забезпечення є платним і не кожна компанія хоче витратити на це гроші - Відсутність аналогічного досвіду
Можливості: – Впровадження нових реалізацій в програмі	Загрози: - Поява реального конкурента на ринку

На основі SWOT-аналізу розробляються альтернативи ринкової поведінки (перелік заходів) для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (таблиця 4.13).

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	Диверсифікація	Створення додаткових можливостей програми, наприклад, супроводжуюча аналітика	6 місяців
	Розвиток ринку	Вихід продукту на міжнародний ринок	2 роки

4.5 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 4.14).

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

п/п	Опис профілю цільової групи потенційних клієнтів	Гото вність споживачів сприйняти продукт	Орієн товний попит в межах цільової групи (сегменту)	Інтенси вність конкуренції в сегменті	Ва жкість входу у сегмент
	Відео платформа	Висока	Високий	Низька	Середня
Які цільові групи обрано: відеоплатформа					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
	Спеціалізація	Зосеред ження на одному сегменті ринку	Науковість підходу, гарантії ефективної роботи	Стратегія спеціалізації

Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

п/п	Чи є проект «першопроходцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
	Так	Так	Ні	Стратегія лідера

На основі вимог споживачів з обраних сегментів до постачальника (стартап-компанії) та до продукту, а також в залежності від обраної базової стратегії розвитку (таблиця 4.15) та стратегії конкурентної поведінки (таблиця 4.16) розробляється стратегія позиціонування (таблиця 4.17). що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати проект.

Таблиця 4.17 – Визначення стратегії позиціонування

п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
	Ефективність; відповідальність; новизна на ринку.	Стратегія спеціалізації	Науковість підходу, гарантії ефективної роботи, перспективи підтримки і подальшого розвитку	Науковість, точність, новизна

4.6 Розроблення маркетингової програми стартап-проекту

Першим кроком є формування маркетингової концепції товару, який отримає споживач. Для цього у таблиці 4.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
	Автоматизація процесів	Реалізація цієї автоматизації	У конкурентів дані генерація персональних рекомендацій вимагає мануальних рішень або ж перетренування моделі
	Аналітичний супровід	Реалізація аналітичного а технічного супроводу	Такого немає у жодного з і так слабких конкурентів

Надалі розробляється трирівнева маркетингова модель товару: уточнюються ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Розумний програмний продукт із високим рівнем точності оцінювання та зручним інтерфейсом.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Функціональність	М	Тх
	2. Швидкодія	М	Тх
	3. Зручність	М	Е
	4. Зовнішній вигляд інтерфейсу	М	Е/Ор
	Якість: продукт має відповідати міжнародним стандартам		
III. Товар із підкріпленням	Використовується реклама		
	Використовуються тимчасові знижки		
За рахунок чого потенційний товар буде захищено від копіювання: за рахунок електронних ключів та інтелектуальної власності.			

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (остаточне визначення ціни відбувається під час фінансово-економічного аналізу проєкту), яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів (таблиця 4.20).

Таблиця 4.20 – Визначення меж встановлення ціни

п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	Немає	Вищий	Високий	Індивідуальна домовленість в залежності від складності інтеграції

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 4.21).

Таблиця 4.21 – Формування системи збуту

п/ п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	Компанія надає свої анонімізовані дані в користування	Результати і аналітичний супровід		

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

п/ п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
	Вимагають інтерпретацію результатів і швидкого реагування у випадку помилок	Телефон, електронна пошта, індивідуальні планові зустрічі	Гарантія результату, швидка підтримка, знижки для постійних клієнтів	Привернути увагу та зацікавити цільових клієнтів.	Надання знижок на перші місяці користування продуктом, а також для постійних клієнтів

4.7 Висновки до розділу

У даному розділі було розроблено проект створення персоналізованих рекомендацій відео “Video Recommendation System”. Для цього було розглянуто та запропоновано потенційні ринкові стратегії та можливості; були описані потенційні конкуренти та переваги і недоліки у порівнянні з ними; були розглянуті бар’єри входження на ринок. Даний проект доцільно розвивати та реалізовувати.

ВИСНОВКИ

Сучасні умови економічної конкуренції вимагають від веб платформ створення нових технологій для підвищення ключових показників ефективності, залучення нових користувачів. В даній магістерській дисертації вирішувалась задача побудови системи для створення відеорекомендацій. В роботі отримано наступні результати.

1. Проаналізовано структуру рекомендаційних систем, розглянуто особливості створення рекомендаційних систем для відео.
2. Проаналізовано основні підходи до побудови рекомендаційних систем, виділено їх основні переваги та недоліки. Запропоновано використовувати гібридний підхід в роботі
3. Побудовано модель GraphSAGE з індуктивністю для прогнозування оцінок користувачами відео контенту, виділено основні критерії якості роботи моделі. Розглянуто можливості врахування гетерогенних даних моделлю GraphSAGE.
4. Проведено аналіз результатів прогнозу моделі.
5. Приведено опис архітектури системи для створення відеорекомендацій на основі побудованої моделі.
6. Виділено можливий напрямок для вдосконалення роботи системи: врахування часового контексту при прогнозуванні оцінок користувачів.

ПЕРЕЛІК ПОСИЛАНЬ

1. What are Recommendation Systems in Machine Learning URL: <https://www.analyticssteps.com/blogs/what-are-recommendation-systems-machine-learning> (Last accessed: 11.12.2020).
2. Introduction to Recommender Systems URL: <https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada> (Last accessed: 11.12.2020).
3. Web Information Retrieval | Vector Space Model URL: <https://www.geeksforgeeks.org/web-information-retrieval-vector-space-model/> (Last accessed: 11.12.2020).
4. A Survey of Collaborative Filtering Techniques URL: <https://www.hindawi.com/journals/aai/2009/421425/> (Last accessed: 11.12.2020).
5. OhMyGraphs: GraphSAGE and inductive representation learning URL: <https://medium.com/analytics-vidhya/ohmygraphs-graphsage-and-inductive-representation-learning-ea26d2835331> (Last accessed: 11.12.2020).
6. Hamilton, Will, Zhitao Ying, and Jure Leskovec. “Inductive representation learning on large graphs.” *31st Conference on Neural Information Processing Systems (NIPS 2017)*. P. 4-6
7. Graph Machine Learning in Genomic Prediction URL: <https://www.kdnuggets.com/2020/06/graph-machine-learning-genomic-prediction.html> (Last accessed: 11.12.2020).
8. Overfitting and Underfitting With Machine Learning Algorithms URL: <https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/>

9. J. Chen and J. Zhu. Stochastic training of graph convolutional networks with variance reduction. *35 th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018*. P.2-3

10. RMSE: Root Mean Square Error. URL: <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/> (Last accessed: 11.12.2020).

11. How to Build a Recommendation System in a Graph Database Using a Latent Factor Model URL: <https://towardsdatascience.com/how-to-build-a-recommendation-system-in-a-graph-database-using-a-latent-factor-model-fa2d142f874> (Last accessed: 11.12.2020).